

H5R_CREATE

[Expand all](#) [Collapse all](#)

- [Jump to ...](#)
- [Summary](#)
- [Description](#)
- [Example](#)
- [Switch language ...](#)
- [C](#)
- [C++](#)
- [FORTRAN](#)
- [JAVA](#)

[Summary](#)
[Description](#)
[Example](#)
[JAVA](#)
[FORTRAN](#)
[C++](#)
[C](#)

H5R_CREATE

Creates a reference

Procedure:

H5R_CREATE (ref, loc_id, name, ref_type, space_id)

Signature:

```
herr_t H5Rcreate(  
    void *ref,  
    hid_t loc_id,  
    const char *name,  
    H5R_type_t ref_type,  
    hid_t space_id  
)
```

Fortran90 Interface: h5rcreate_f

To create an object reference:

Signature:

```
SUBROUTINE h5rcreate_f(loc_id, name, ref, hdferr)
  INTEGER(HID_T)      , INTENT(IN)      :: loc_id
  CHARACTER(LEN=*)    , INTENT(IN)      :: name
  TYPE(hobj_ref_t_f)  , INTENT(INOUT)   :: ref
  INTEGER              , INTENT(OUT)    :: hdferr
```

Inputs:

loc_id - Location identifier
name - Name of the object at location specified by loc_id identifier

Outputs:

ref - Object reference
hdferr - Error code:
0 on success and -1 on failure

To create a region reference:

Signature:

```
SUBROUTINE h5rcreate_f(loc_id, name, space_id, ref, hdferr)
  INTEGER(HID_T)      , INTENT(IN)      :: loc_id
  CHARACTER(LEN=*)    , INTENT(IN)      :: name
  INTEGER(HID_T)      , INTENT(IN)      :: space_id
  TYPE(hdset_reg_ref_t_f), INTENT(OUT)   :: ref
  INTEGER              , INTENT(OUT)    :: hdferr
```

Inputs:

loc_id - Location identifier
name - Name of the dataset at location specified by loc_id identifier
space_id - Dataset's dataspace identifier

Outputs:

ref - Dataset region reference
hdferr - Error code
0 on success and -1 on failure

Fortran2003 Interface: h5rcreate_f

Signature:

```
SUBROUTINE h5rcreate_f(loc_id, name, ref_type, ref, hdferr, space_id)
  INTEGER(HID_T)      , INTENT(IN)      :: loc_id
  CHARACTER(LEN=*)    , INTENT(IN)      :: name
  INTEGER              , INTENT(IN)      :: ref_type
  TYPE(C_PTR)         , INTENT(INOUT)    :: ref
  INTEGER              , INTENT(OUT)     :: hdferr
  INTEGER(HID_T)      , INTENT(IN)      , OPTIONAL :: space_id
```

Inputs:

loc_id - Location identifier used to locate the object being pointed to.
name - Name of object at location loc_id.
ref_type - Type of reference:
H5R_OBJECT
H5T_STD_REF_DSETREG

Outputs:

ref - Reference created by the function call.
hdferr - Error code
0 on success and -1 on failure

Optional Parameter:

space_id - Dataspace identifier that describes selected region.

Parameters:

<i>void *ref</i>	OUT: Reference created by the function call
<i>hid_t loc_id</i>	IN: Location identifier used to locate the object being pointed to
<i>const char *name</i>	IN: Name of object at location <i>loc_id</i>
<i>H5R_type_t ref_type</i>	IN: Type of reference
<i>hid_t space_id</i>	IN: Dataspace identifier with selection. Used only for dataset region references; pass as -1 if reference is an object reference, i.e., of type H5R_OBJECT

Description:

H5R_CREATE creates the reference, *ref*, of the type specified in *ref_type*, pointing to the object *name* located at *loc_id*.

The HDF5 library maps the *void* type specified above for *ref* to the type specified in *ref_type*, which will be one of those appearing in the first column of the following table. The second column of the table lists the HDF5 constant associated with each reference type.

<i>hdset_reg_ref_t</i>	H5R_DATASET_REGION	Dataset region reference
<i>hobj_ref_t</i>	H5R_OBJECT	Object reference

The parameters *loc_id* and *name* are used to locate the object.

The parameter *space_id* identifies the dataset region that a dataset region reference points to. This parameter is used only with dataset region references and should be set to -1 if the reference is an object reference, H5R_OBJECT.

Returns:

Returns a non-negative value if successful; otherwise returns a negative value.

Example:

```
1_10 / C / H5T / h5ex_t_objref.c [56:62]          master   HDFFV/hdf5-exam
ples
/*
 * Create references to the previously created objects.  Passing -1
 * as space_id causes this parameter to be ignored.  Other values
 * besides valid dataspace result in an error.
 */
status = H5Rcreate (&wdata[0], file, "G1", H5R_OBJECT, -1);
status = H5Rcreate (&wdata[1], file, "DS2", H5R_OBJECT, -1);
```

DFFV/hdf5-examples

```
! Create references to the previously created objects. note, space_id
! is not needed for object references.
!
f_ptr = C_LOC(wdata(1))
CALL H5Rcreate_f(file, "G1", H5R_OBJECT_F, f_ptr, hdferr)
f_ptr = C_LOC(wdata(2))
CALL H5Rcreate_f(file, "DS2", H5R_OBJECT_F, f_ptr, hdferr)
```

History:

Release	Change
1.8.8	Fortran updated to Fortran2003.
1.8.0	C function introduced in this release.

--- Last Modified: May 03, 2019 | 01:19 PM